SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

SPYWOLF

Security Audit Report

bworker

Completed on

June 27, 2022

MADE IN USA 🇺🇸

OVERVIEW

This audit has been prepared for **bWorker** to review the main aspects of the project to help investors make make an informative decision during their research process.

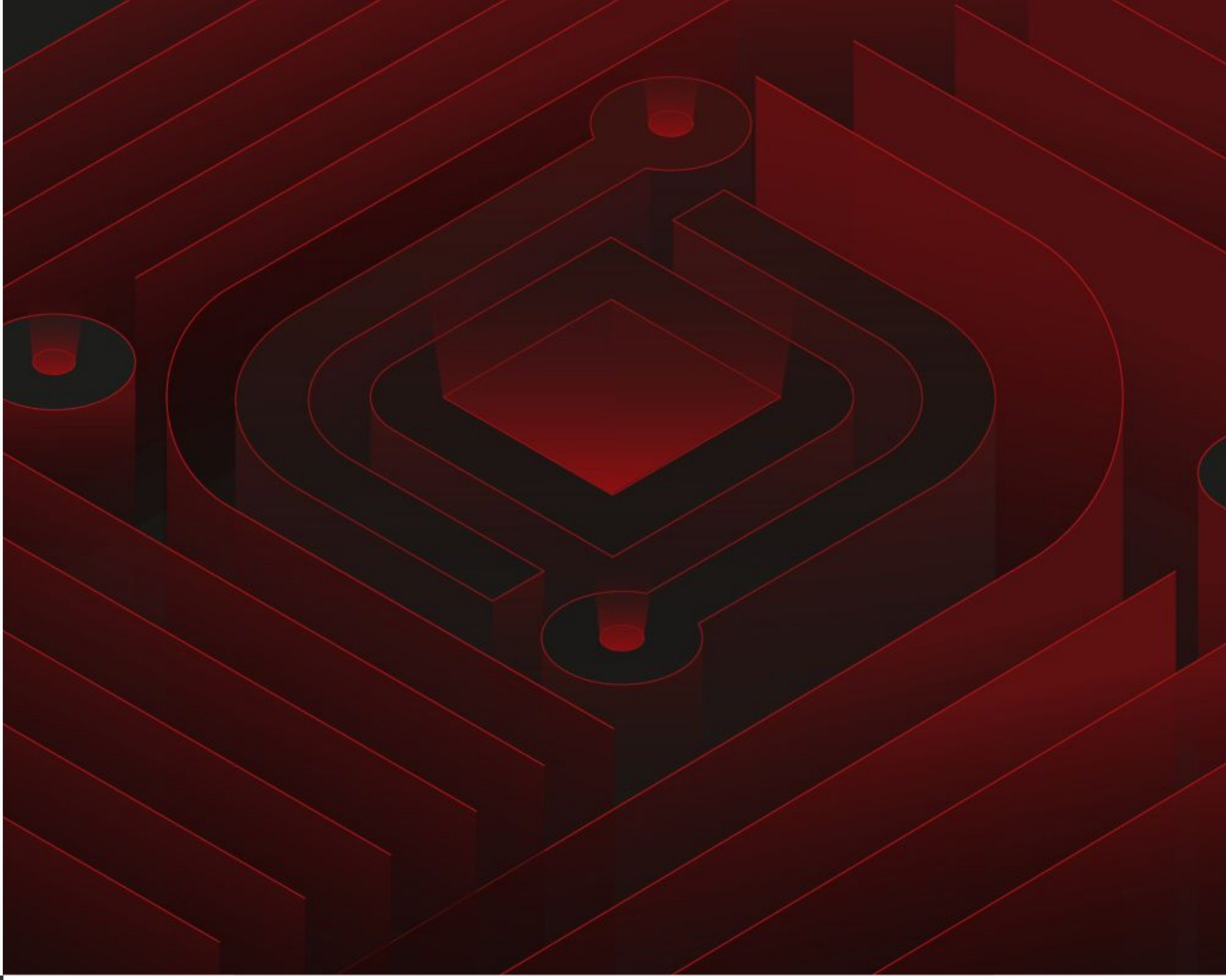
You will find a a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Owners' wallets
- ✓ Tokenomics
- ✓ Team transparency and goals
- ✓ Website's age, code, security and UX
- ✓ Whitepaper and roadmap
- ✓ Social media & online presence

> *The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal*
>
> - SPYWOLF Team -

TABLE OF CONTENTS

bWorker



PROJECT DESCRIPTION

According to their whitepaper:
Bworker is defi protocol in which users can earn up to 15% passive income daily. The native token of the protocol will be $BWP.
Users can join by owning a NFTs that will worth 100 $BUSD - 500 $BUSD.
By owning NFT, the system will automatically reward 3% - 15% daily in $BWP.
When users mint their NFTs they can participate in scibo-like games.
Users can withdraw daily rewards $BWP only by winning those games.
If users win, they can withdraw 10% of their winning account per day.

Release Date: Presale starts on June, 2022

Category: NFT

CONTRACT INFO

Token Name
BWorker

Symbol
BWP

Contract Address
0xE682A56ACB194D2aFA06BbA838D259B0C888666F

Network
Binance Smart Chain

Language
Solidity

Deployment Date
June 23, 2022

Verified?
Yes

Total Supply
1,000,000

Status
Not launched

TAXES

Buy Tax
none

Sell Tax
none

Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of June 27, 2022)

Status:

Not Launched!

Liquidity

Not added yet

Burn

No burnt tokens

MaxTxAmount

1,500

Additional info

LP Address(es)

Liquidity not added yet

Legend NFT price:
500 BUSD

NFT price:
100 BUSD

Legend NFTs can be sold:
115792089237316195423570985008687907853269984665640564039457584007913129639935

NFTs can be sold:
115792089237316195423570985008687907853269984665640564039457584007913129639935

TOKEN TRANSFERS STATS

Transfer Count	1
Uniq Senders	1
Uniq Receivers	1
Total Amount	1 BWP
Median Transfer Amount	1 BWP
Average Transfer Amount	1 BWP
First transfer date	2022-06-24
Last transfer date	2022-06-24
Days token transferred	1

SMART CONTRACT STATS

Calls Count	2
External calls	2
Internal calls	0
Transactions count	2
Uniq Callers	2
Days contract called	2
Last transaction time	2022-06-24 03:15:29 UTC
Created	2022-06-23 15:28:10 UTC
Create TX	0xa7a7b289f62d2fd8b057cea293e32b35986b4e2be3abb66604b3917883b553c3
Creator	0x666637c68230fd62b0ee0a47f2b9a6b5193adddd

FEATURED WALLETS

Owner address	0x666637c68230fd62b0ee0a47f2b9a6b5193adddd
Treasury receiver	0x5555F52F6765FF03554c821A262Eb585C5D20000
Insurance receiver	0x77771d92ecea6ba92534207528a7c6eaea252222
Buyback receiver	0x00008349d1efab455b857ecc57b626c556ab2222
Add liquidity receiver	0x00008349d1efab455b857ecc57b626c556ab2222
LP address	**No liquidity added yet**

TOP 3 UNLOCKED WALLETS

1	100%	Same as treasury receiver

⚠️Tokens are not distributed yet

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

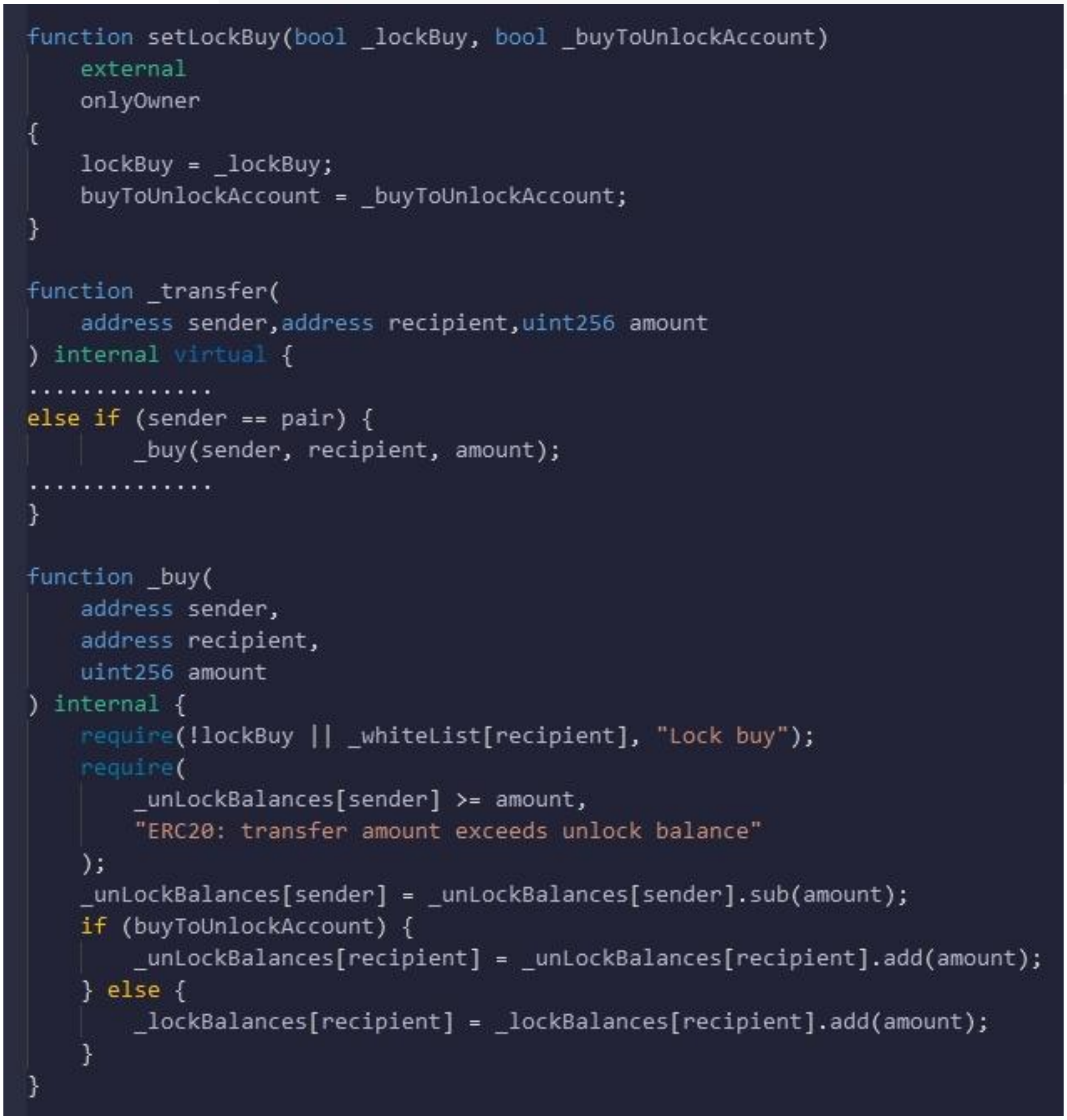
Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can disable buys if lockBuy variable is set to true.
Owner can restrict users to sell the bought amount if buyToUnlockAccount is set to false - the amount bought is assigned to the holder's lockBalances (which can be used in their NFT games).

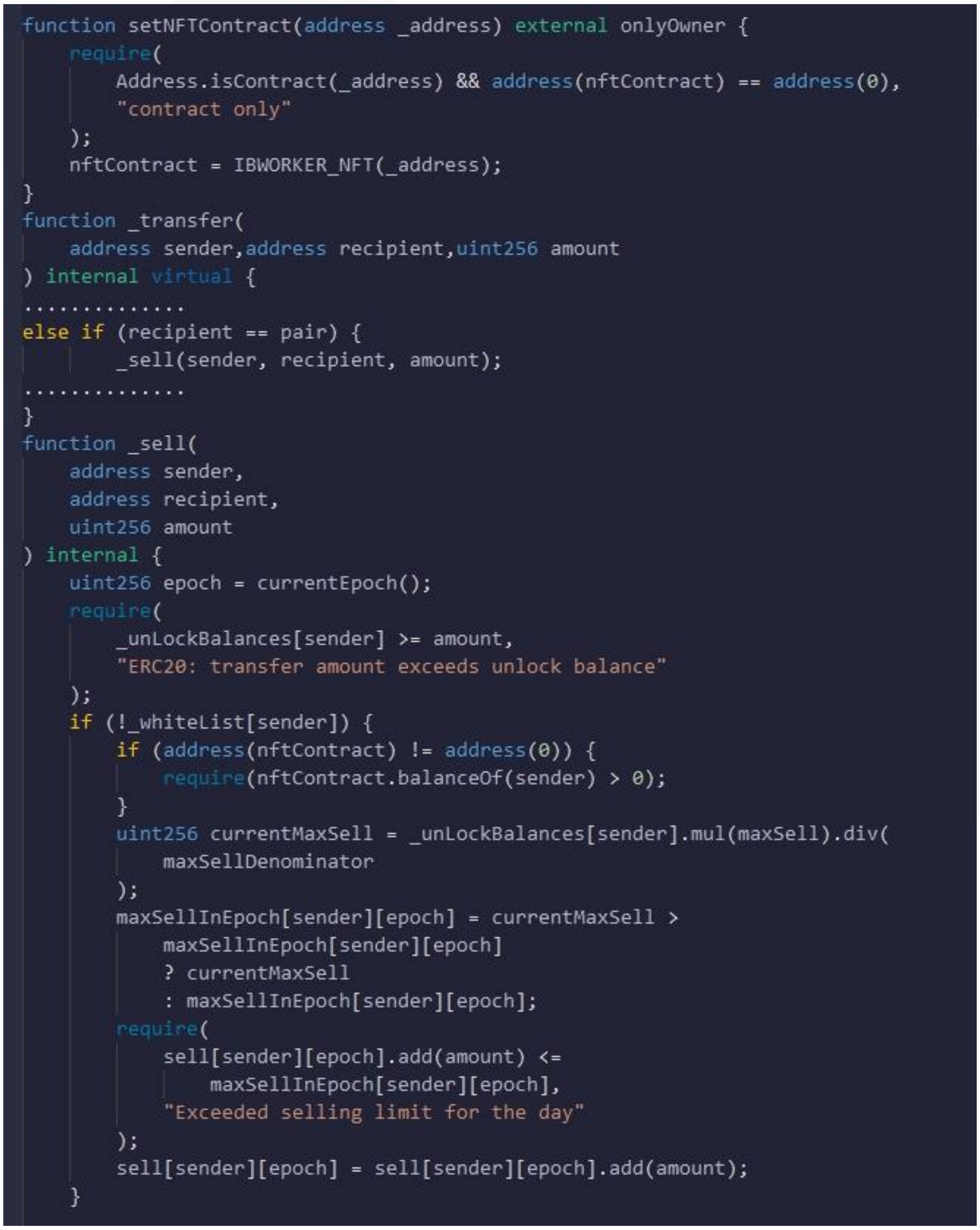


```
function setLockBuy(bool _lockBuy, bool _buyToUnlockAccount)
    external
    onlyOwner
{
    lockBuy = _lockBuy;
    buyToUnlockAccount = _buyToUnlockAccount;
}

function _transfer(
    address sender,address recipient,uint256 amount
) internal virtual {
..............
else if (sender == pair) {
        _buy(sender, recipient, amount);
..............
}

function _buy(
    address sender,
    address recipient,
    uint256 amount
) internal {
    require(!lockBuy || _whiteList[recipient], "Lock buy");
    require(
        _unLockBalances[sender] >= amount,
        "ERC20: transfer amount exceeds unlock balance"
    );
    _unLockBalances[sender] = _unLockBalances[sender].sub(amount);
    if (buyToUnlockAccount) {
        _unLockBalances[recipient] = _unLockBalances[recipient].add(amount);
    } else {
        _lockBalances[recipient] = _lockBalances[recipient].add(amount);
    }
}
```

FOUND THREATS

High Risk

When the NFT contract is assigned, users must hold NFT in order to be able to sell their tokens.
Currently there is no NFT contract assigned and the users can sell the bought unlocked tokens up to maxSell amount without NFT holdings.
Logic of future NFT contract is out of scope of the current audit.

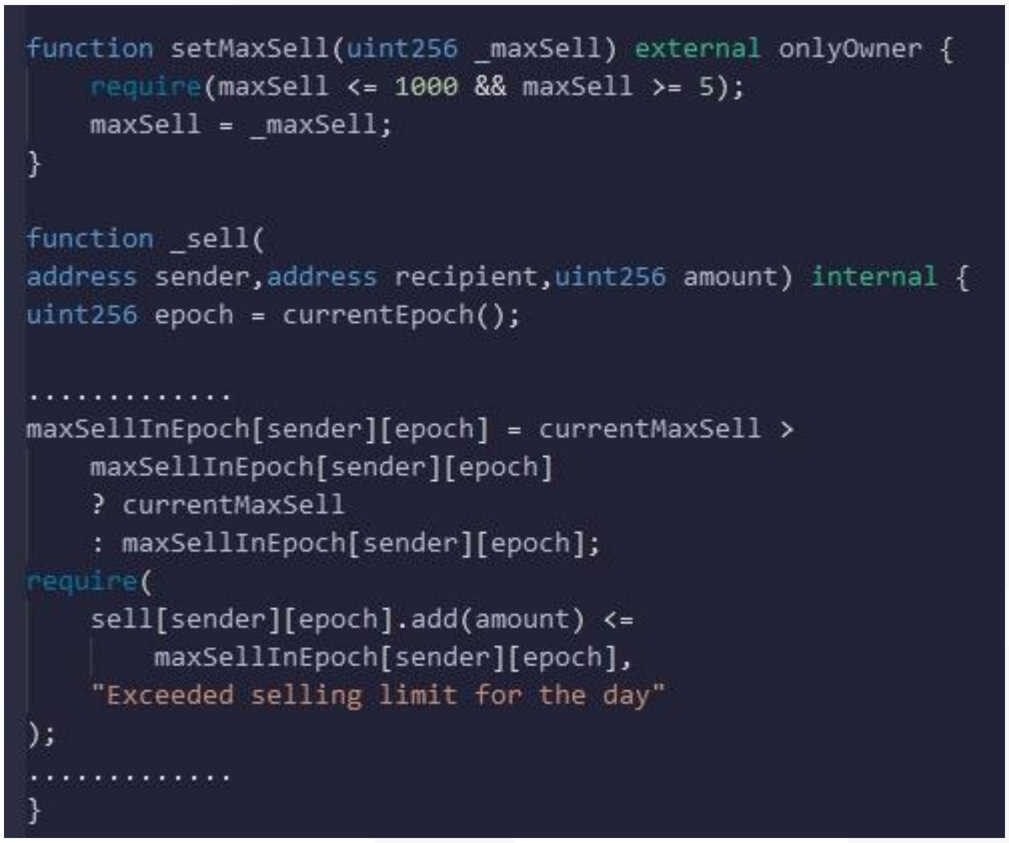


```
function setNFTContract(address _address) external onlyOwner {
    require(
        Address.isContract(_address) && address(nftContract) == address(0),
        "contract only"
    );
    nftContract = IBWORKER_NFT(_address);
}
function _transfer(
    address sender,address recipient,uint256 amount
) internal virtual {
..............
else if (recipient == pair) {
        _sell(sender, recipient, amount);
..............
}
function _sell(
    address sender,
    address recipient,
    uint256 amount
) internal {
    uint256 epoch = currentEpoch();
    require(
        _unLockBalances[sender] >= amount,
        "ERC20: transfer amount exceeds unlock balance"
    );
    if (!_whiteList[sender]) {
        if (address(nftContract) != address(0)) {
            require(nftContract.balanceOf(sender) > 0);
        }
        uint256 currentMaxSell = _unLockBalances[sender].mul(maxSell).div(
            maxSellDenominator
        );
        maxSellInEpoch[sender][epoch] = currentMaxSell >
            maxSellInEpoch[sender][epoch]
            ? currentMaxSell
            : maxSellInEpoch[sender][epoch];
        require(
            sell[sender][epoch].add(amount) <=
                maxSellInEpoch[sender][epoch],
            "Exceeded selling limit for the day"
        );
        sell[sender][epoch] = sell[sender][epoch].add(amount);
    }
```

FOUND THREATS

⚠ High Risk

Owner can set max amount that holder can sell for 1 day (epoch), but can't lower it than 0.5% of holder's current balance.
For example if you have 100 tokens and this is set to 0.5%, you can sell only 0.5 tokens per epoch.

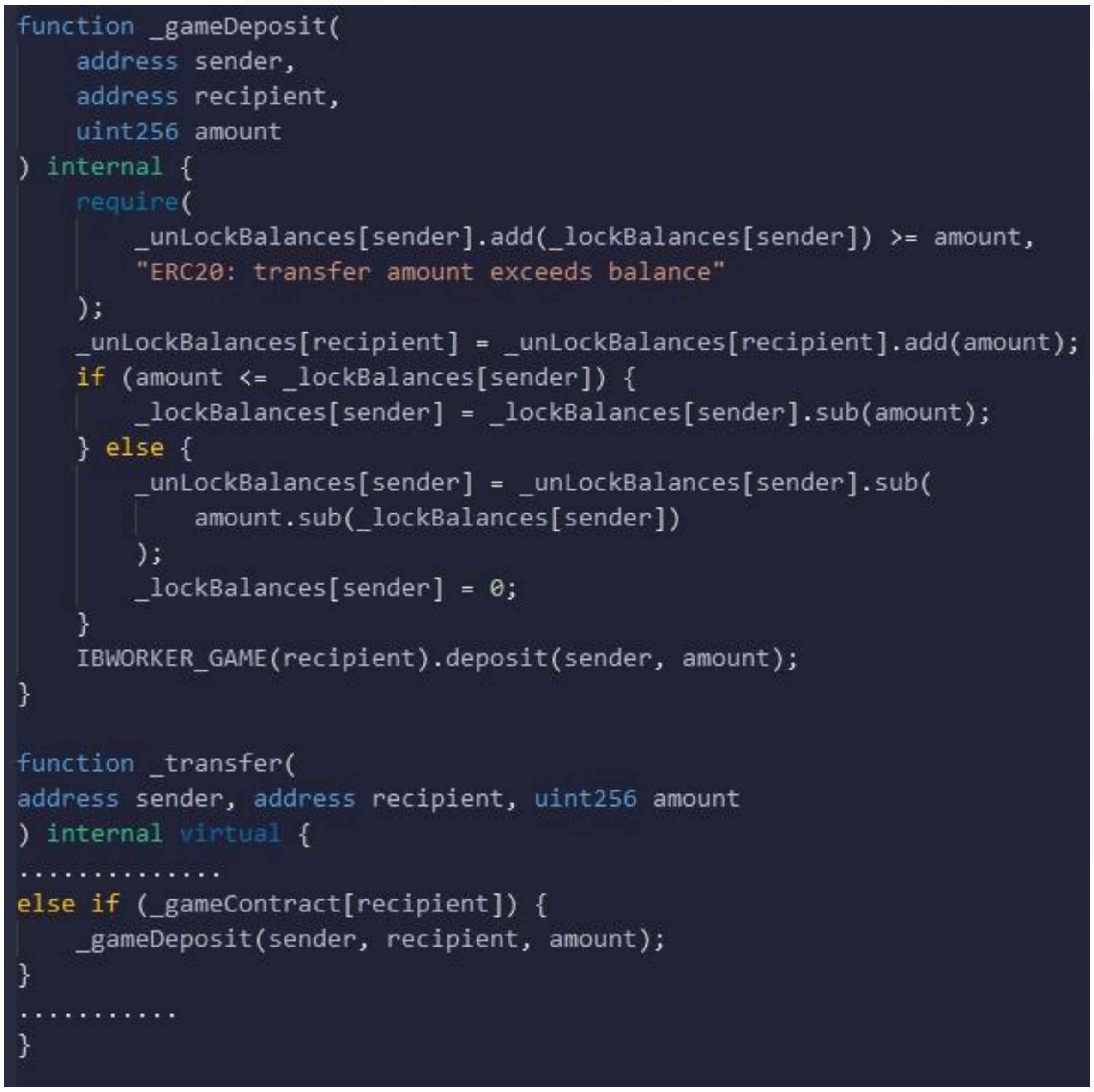


```
function setMaxSell(uint256 _maxSell) external onlyOwner {
    require(maxSell <= 1000 && maxSell >= 5);
    maxSell = _maxSell;
}

function _sell(
address sender,address recipient,uint256 amount) internal {
uint256 epoch = currentEpoch();

.............
maxSellInEpoch[sender][epoch] = currentMaxSell >
    maxSellInEpoch[sender][epoch]
    ? currentMaxSell
    : maxSellInEpoch[sender][epoch];
require(
    sell[sender][epoch].add(amount) <=
        maxSellInEpoch[sender][epoch],
    "Exceeded selling limit for the day"
);
.............
}
```

FOUND THREATS

⚠ High Risk

Deposits in the game contract may fail and deduct unfair tokens amount from holders balances.
If amount that sender deposit to the game contract is above the _lockBalances of sender it will deduct the whole _lockBalances of the sender + the amount sent.



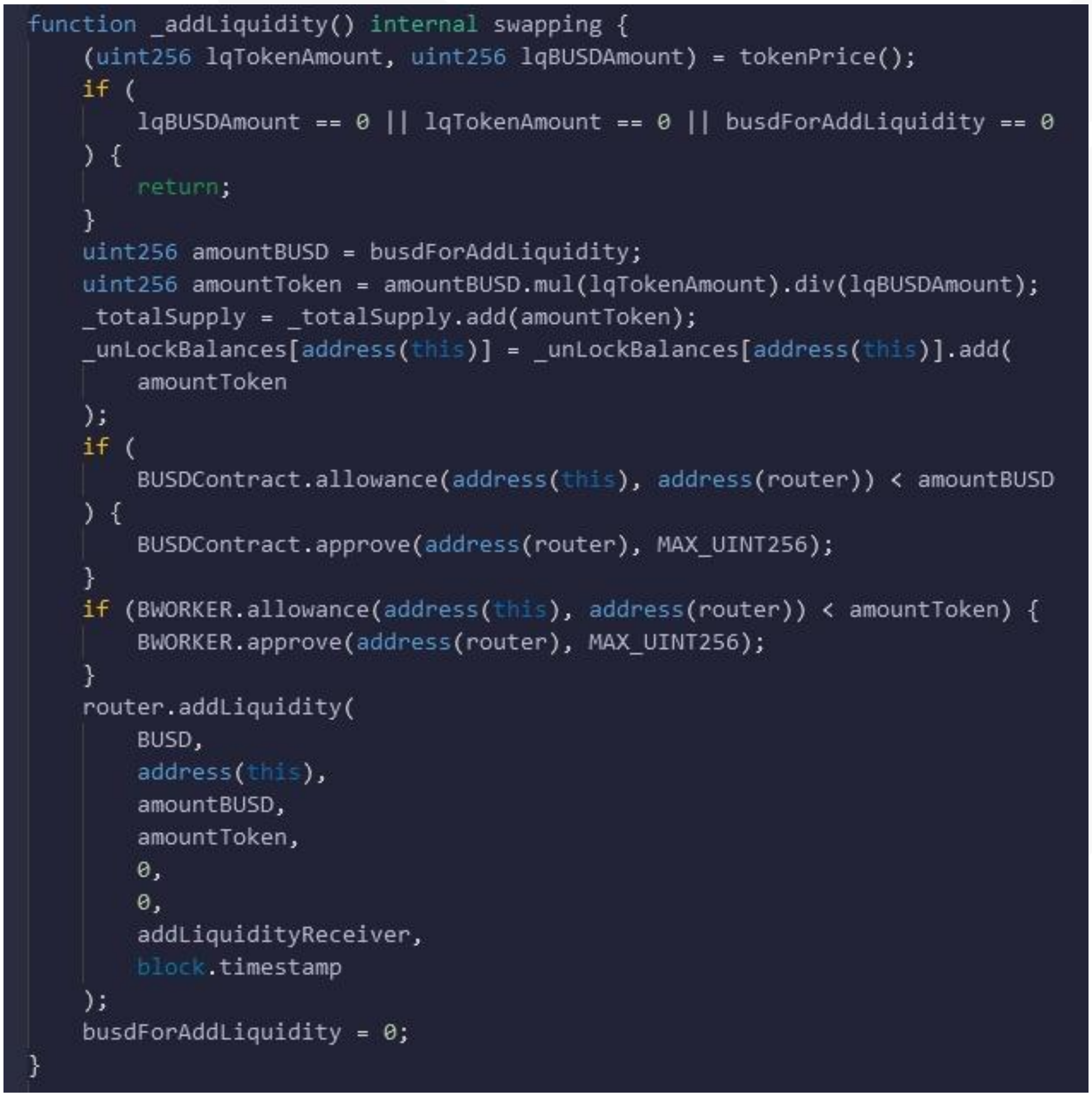
```
function _gameDeposit(
    address sender,
    address recipient,
    uint256 amount
) internal {
    require(
        _unLockBalances[sender].add(_lockBalances[sender]) >= amount,
        "ERC20: transfer amount exceeds balance"
    );
    _unLockBalances[recipient] = _unLockBalances[recipient].add(amount);
    if (amount <= _lockBalances[sender]) {
        _lockBalances[sender] = _lockBalances[sender].sub(amount);
    } else {
        _unLockBalances[sender] = _unLockBalances[sender].sub(
            amount.sub(_lockBalances[sender])
        );
        _lockBalances[sender] = 0;
    }
    IBWORKER_GAME(recipient).deposit(sender, amount);
}

function _transfer(
address sender, address recipient, uint256 amount
) internal virtual {
..............
else if (_gameContract[recipient]) {
    _gameDeposit(sender, recipient, amount);
}
...........
}
```

Note:
Lets take the following scenario into consideration:
The sender have _lockBalances = 100 and _unLockBalances = 200.
The sender want to deposit 150 tokens into the game contact.
This means that from holder's current _unLockBalances will be deducted the deposited amount combined with holder's _unLockBalances.
So it becomes 200 - 150 - 100 = -50.
So the holder wants to deposit 150 tokens into the contract but is deducted with 250 tokens (if the tx succeeds).

FOUND THREATS

⚠ High Risk

Autoliquidity add function mints new token supply and sends it to the addLiquidityReceiver (private wallet) with the accumulated BUSD from the minted NFTs.

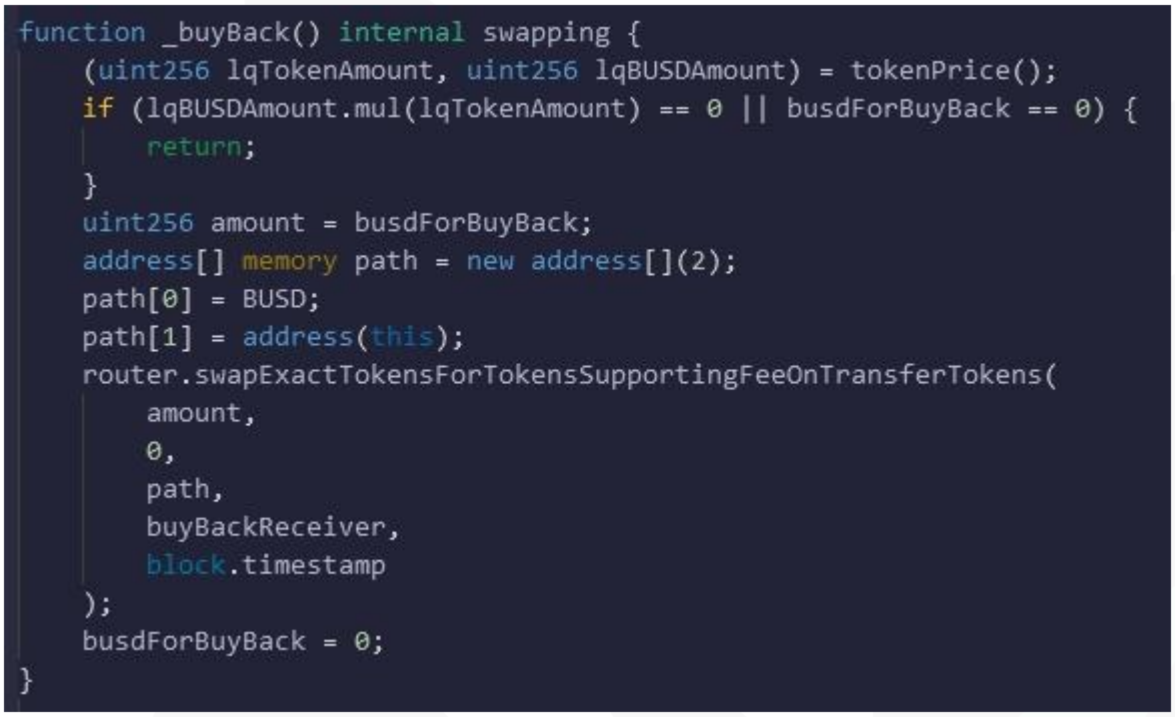


```
function _addLiquidity() internal swapping {
    (uint256 lqTokenAmount, uint256 lqBUSDAmount) = tokenPrice();
    if (
        lqBUSDAmount == 0 || lqTokenAmount == 0 || busdForAddLiquidity == 0
    ) {
        return;
    }
    uint256 amountBUSD = busdForAddLiquidity;
    uint256 amountToken = amountBUSD.mul(lqTokenAmount).div(lqBUSDAmount);
    _totalSupply = _totalSupply.add(amountToken);
    _unLockBalances[address(this)] = _unLockBalances[address(this)].add(
        amountToken
    );
    if (
        BUSDContract.allowance(address(this), address(router)) < amountBUSD
    ) {
        BUSDContract.approve(address(router), MAX_UINT256);
    }
    if (BWORKER.allowance(address(this), address(router)) < amountToken) {
        BWORKER.approve(address(router), MAX_UINT256);
    }
    router.addLiquidity(
        BUSD,
        address(this),
        amountBUSD,
        amountToken,
        0,
        0,
        addLiquidityReceiver,
        block.timestamp
    );
    busdForAddLiquidity = 0;
}
```

FOUND THREATS

⚠ High Risk

When buyback() function is triggered, the buyback receiver (private wallet) receives the buyback tokens.

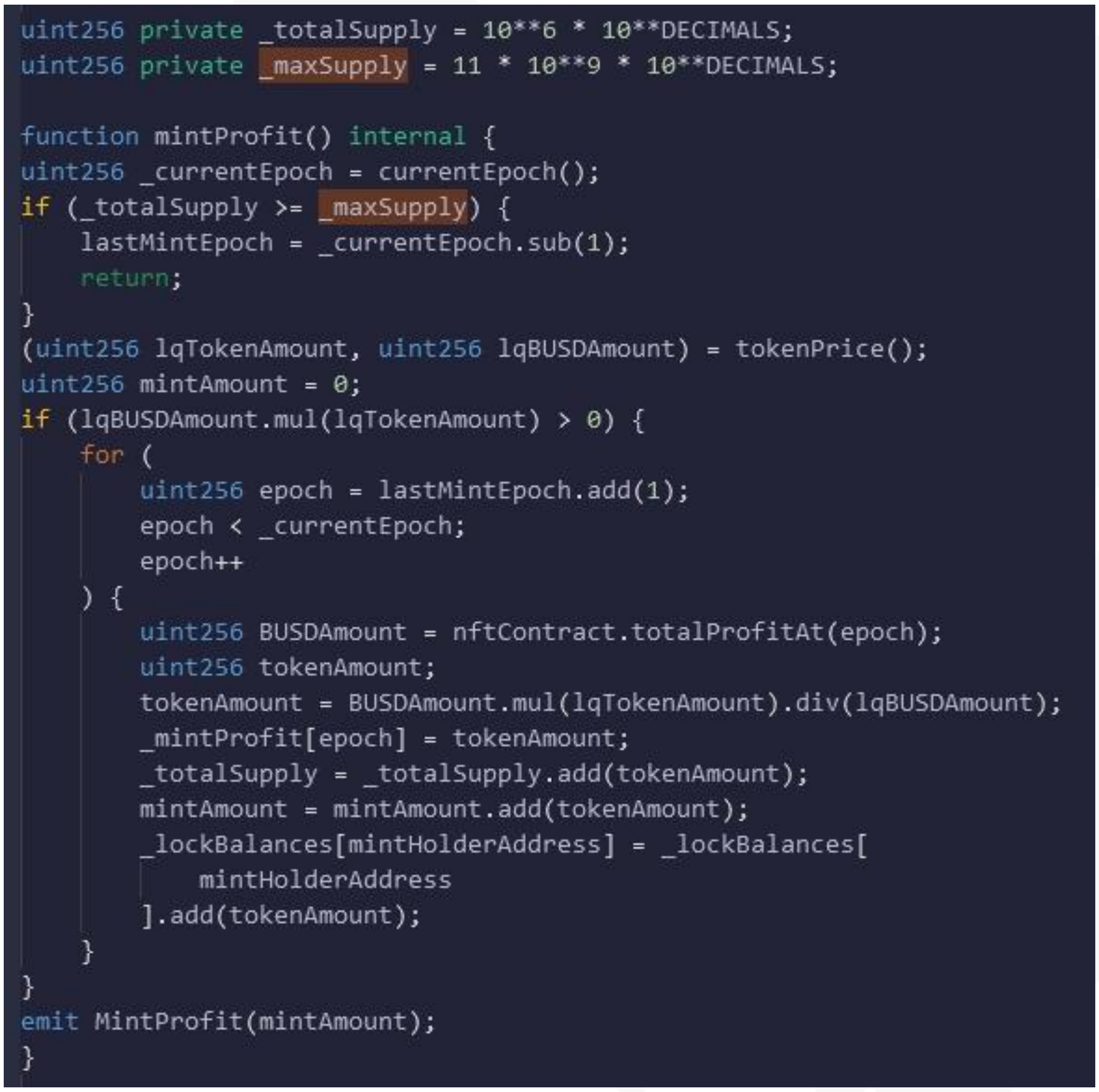


```
function _buyBack() internal swapping {
    (uint256 lqTokenAmount, uint256 lqBUSDAmount) = tokenPrice();
    if (lqBUSDAmount.mul(lqTokenAmount) == 0 || busdForBuyBack == 0) {
        return;
    }
    uint256 amount = busdForBuyBack;
    address[] memory path = new address[](2);
    path[0] = BUSD;
    path[1] = address(this);
    router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
        amount,
        0,
        path,
        buyBackReceiver,
        block.timestamp
    );
    busdForBuyBack = 0;
}
```

FOUND THREATS

High Risk

This is rebase token with _maxSupply up to 11,000,000,000.
Current supply is 1,000,000.
Rebase tokens can lead to significant token price inflation over time.

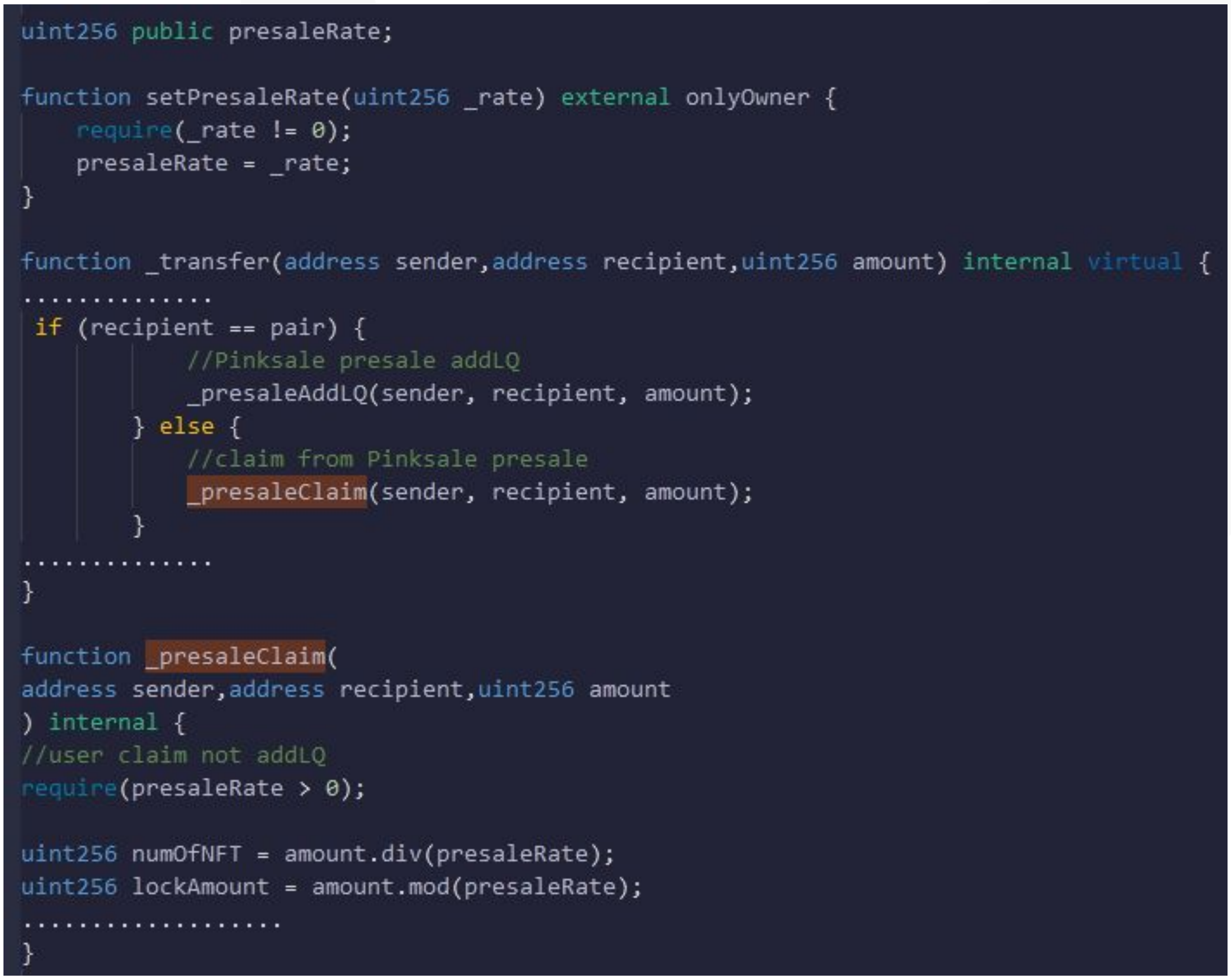


```
uint256 private _totalSupply = 10**6 * 10**DECIMALS;
uint256 private _maxSupply = 11 * 10**9 * 10**DECIMALS;

function mintProfit() internal {
uint256 _currentEpoch = currentEpoch();
if (_totalSupply >= _maxSupply) {
    lastMintEpoch = _currentEpoch.sub(1);
    return;
}
(uint256 lqTokenAmount, uint256 lqBUSDAmount) = tokenPrice();
uint256 mintAmount = 0;
if (lqBUSDAmount.mul(lqTokenAmount) > 0) {
    for (
        uint256 epoch = lastMintEpoch.add(1);
        epoch < _currentEpoch;
        epoch++
    ) {
        uint256 BUSDAmount = nftContract.totalProfitAt(epoch);
        uint256 tokenAmount;
        tokenAmount = BUSDAmount.mul(lqTokenAmount).div(lqBUSDAmount);
        _mintProfit[epoch] = tokenAmount;
        _totalSupply = _totalSupply.add(tokenAmount);
        mintAmount = mintAmount.add(tokenAmount);
        _lockBalances[mintHolderAddress] = _lockBalances[
            mintHolderAddress
        ].add(tokenAmount);
    }
}
emit MintProfit(mintAmount);
}
```

FOUND THREATS

⚠ High Risk

Owner can change presale rate.
Users number of NFTs received and locked tockens received will be based on the presale rate.
Tokens from presale are impossible to claim if the owner don't change the presaleRate above 0.
Once it is above 0 it cannot be set to 0 again.
Current presaleRate is 0.

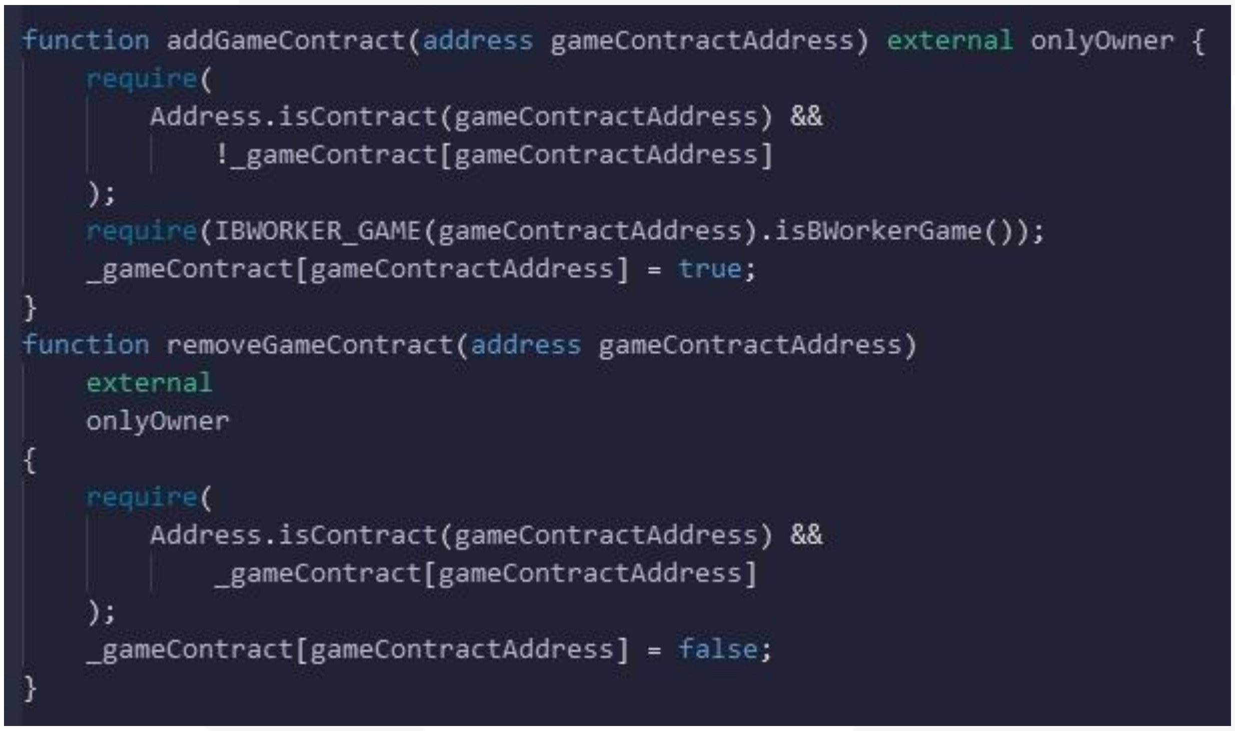


```
uint256 public presaleRate;

function setPresaleRate(uint256 _rate) external onlyOwner {
    require(_rate != 0);
    presaleRate = _rate;
}

function _transfer(address sender,address recipient,uint256 amount) internal virtual {
..............
 if (recipient == pair) {
           //Pinksale presale addLQ
           _presaleAddLQ(sender, recipient, amount);
        } else {
           //claim from Pinksale presale
           _presaleClaim(sender, recipient, amount);
        }
..............
}

function _presaleClaim(
address sender,address recipient,uint256 amount
) internal {
//user claim not addLQ
require(presaleRate > 0);

uint256 numOfNFT = amount.div(presaleRate);
uint256 lockAmount = amount.mod(presaleRate);
...................
}
```

FOUND THREATS

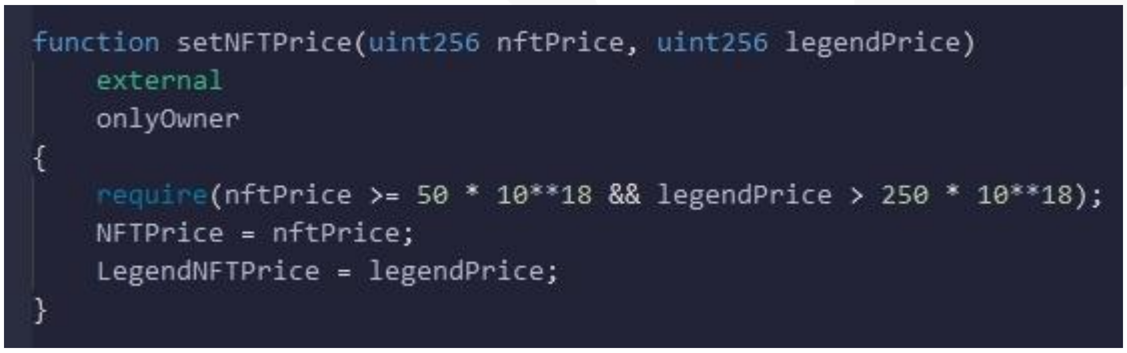
⚠ Medium Risk

Owner can change the game contract which can apply new rules for the games with every new contract set.
There is no assigned game contract at this moment.
Logic of future game contracts is out of scope of current audit.



```
function addGameContract(address gameContractAddress) external onlyOwner {
    require(
        Address.isContract(gameContractAddress) &&
            !_gameContract[gameContractAddress]
    );
    require(IBWORKER_GAME(gameContractAddress).isBWorkerGame());
    _gameContract[gameContractAddress] = true;
}
function removeGameContract(address gameContractAddress)
    external
    onlyOwner
{
    require(
        Address.isContract(gameContractAddress) &&
            _gameContract[gameContractAddress]
    );
    _gameContract[gameContractAddress] = false;
}
```

Owner can change each NFT and Legend NFT price, but can the price below 50 BUSD for NFT and 250 BUSD for Legend NFT.

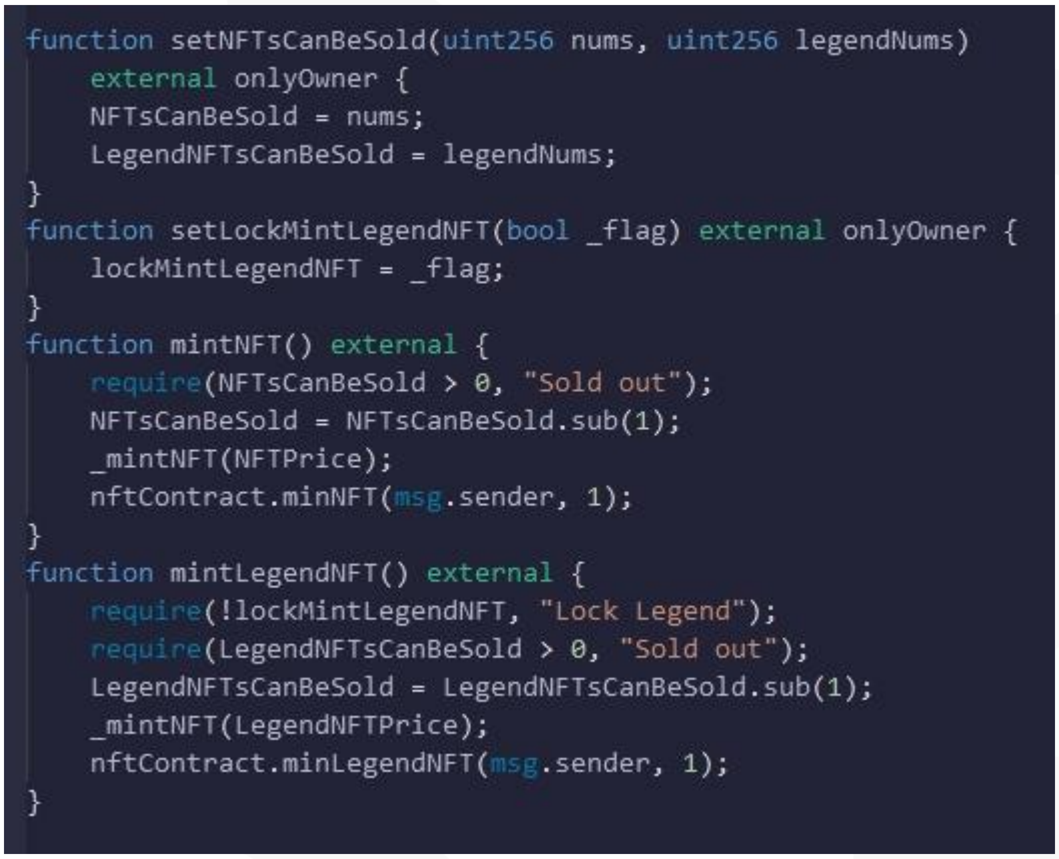


```
function setNFTPrice(uint256 nftPrice, uint256 legendPrice)
    external
    onlyOwner
{
    require(nftPrice >= 50 * 10**18 && legendPrice > 250 * 10**18);
    NFTPrice = nftPrice;
    LegendNFTPrice = legendPrice;
}
```

FOUND THREATS

⚠ Medium Risk

Owner can change total NFTs and Legend NFTs that can be minted and forbid NFT minting.



```
function setNFTsCanBeSold(uint256 nums, uint256 legendNums)
    external onlyOwner {
    NFTsCanBeSold = nums;
    LegendNFTsCanBeSold = legendNums;
}
function setLockMintLegendNFT(bool _flag) external onlyOwner {
    lockMintLegendNFT = _flag;
}
function mintNFT() external {
    require(NFTsCanBeSold > 0, "Sold out");
    NFTsCanBeSold = NFTsCanBeSold.sub(1);
    _mintNFT(NFTPrice);
    nftContract.minNFT(msg.sender, 1);
}
function mintLegendNFT() external {
    require(!lockMintLegendNFT, "Lock Legend");
    require(LegendNFTsCanBeSold > 0, "Sold out");
    LegendNFTsCanBeSold = LegendNFTsCanBeSold.sub(1);
    _mintNFT(LegendNFTPrice);
    nftContract.minLegendNFT(msg.sender, 1);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1 Consider fundamental tradeoffs

2 Be attentive to blockchain properties

3 Ensure careful rollouts

4 Keep contracts simple

5 Stay up to date and track development

bWorker

GOOD PRACTICES FOUND

✓ The smart contract utilizes "SafeMath" to prevent overflows

TOKENOMICS

⚠️There is no information about the initial tokens distribution based on the project's whitepaper and/or website.
100% of the funds spend from investors on NFTs will be collected by the project’s owners based on data collected from the contract.

THE TEAM

The team has privately doxxed to SPYWOLF by completing the following KYC requirements:

- ID Verification
- Video statement
- Video interview with devs
- Owner's wallet verification

KYC INFORMATION

Issuer

SPYWOLF

Members KYC'd

👤

KYC Date

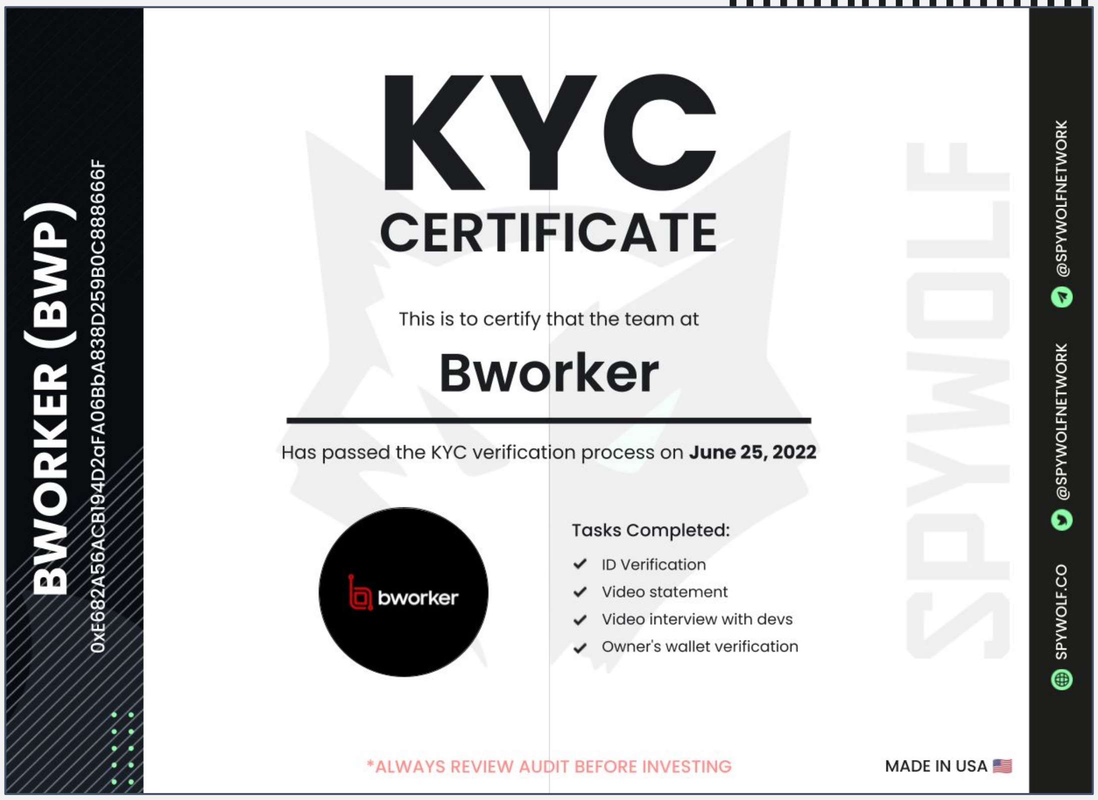
June 25, 2022

Format

Image

Certificate Link

https://github.com/SpyWolfNetwork/KYCs/blob/main/june/KYC_BWORKER_0xE682A56ACB194D2aFA06BbA838D259B0C888666F.png



BWORKER (BWP)

0xE682A56ACB194D2aFA06BbA838D259B0C888666F

KYC CERTIFICATE

This is to certify that the team at

Bworker

Has passed the KYC verification process on **June 25, 2022**

Tasks Completed:

- ✓ ID Verification
- ✓ Video statement
- ✓ Video interview with devs
- ✓ Owner's wallet verification

*ALWAYS REVIEW AUDIT BEFORE INVESTING

MADE IN USA 🇺🇸

bworker
SPYWOLF
@SPYWOLFNETWORK
@SPYWOLFNETWORK
SPYWOLF.CO

WEBSITE

Website URL
https://bworker.io/

Domain Registry
http://www.namesilo.com

Domain Expiration
2023-06-08

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

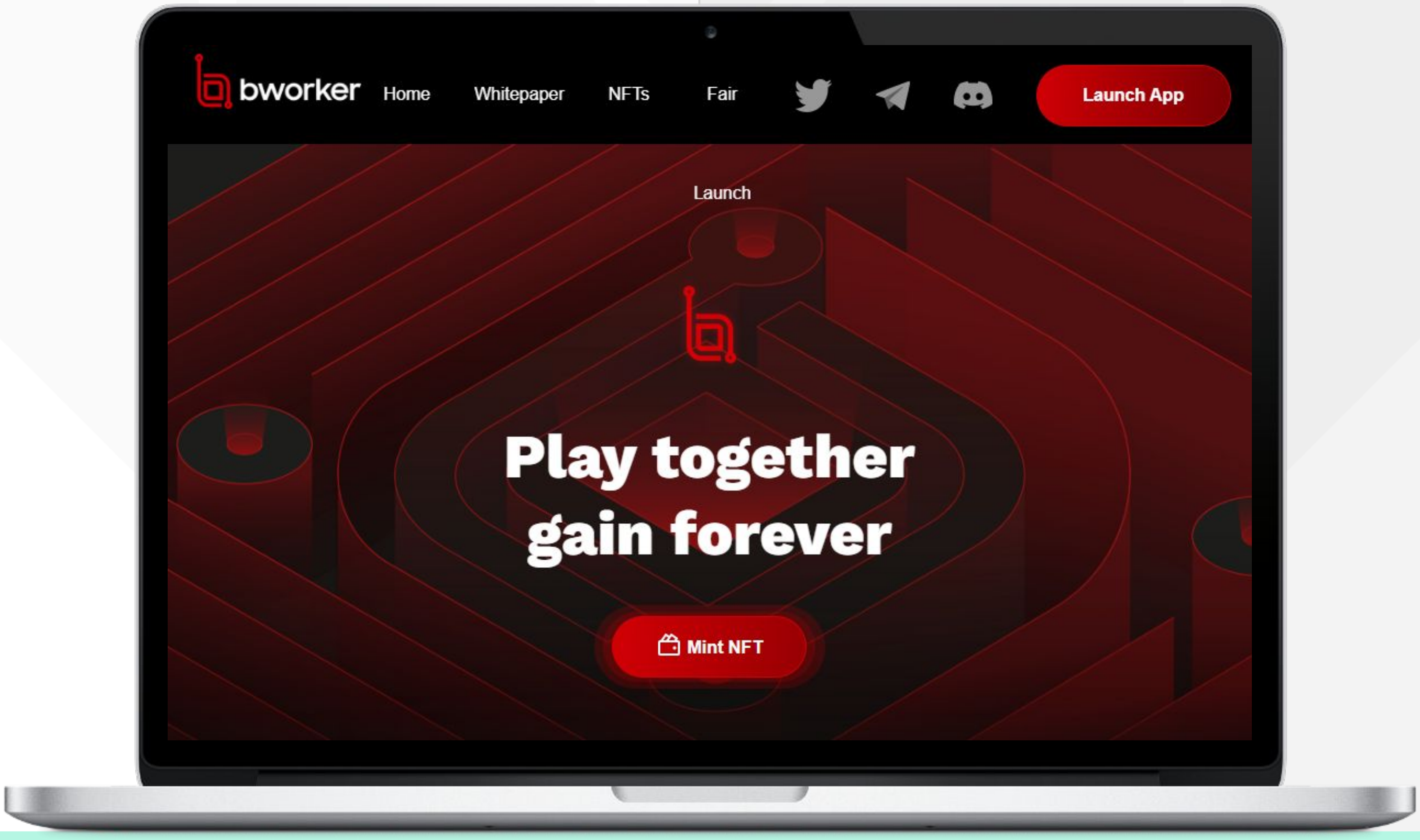
Design
Nice color scheme and overall layout.

Content
The information helps new investors understand what the product does right away. No grammar mistakes found.

Whitepaper
Explanatory but a bit short, few semantic mistakes.

Roadmap
No ⚠️

Mobile-friendly?
Yes



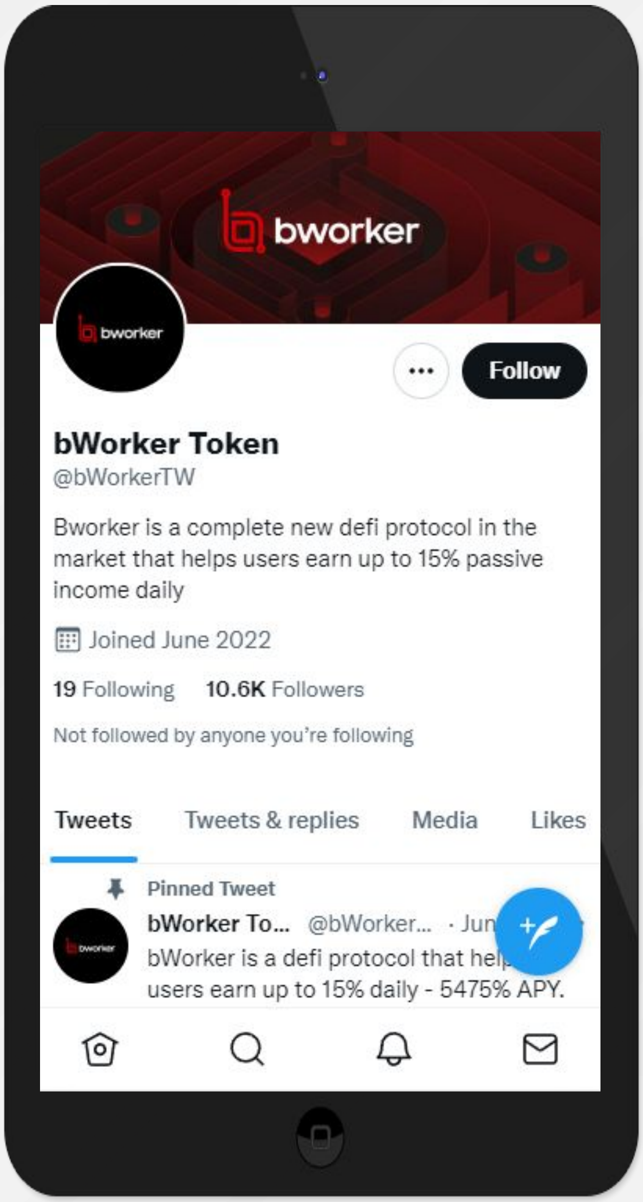
bworker.io

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media presence is relatively new (few days old). Mods are active. The number of bot members may represent a Red Flag.



bWorker Token
@bWorkerTW

Bworker is a complete new defi protocol in the market that helps users earn up to 15% passive income daily

Joined June 2022

19 Following 10.6K Followers

Not followed by anyone you're following

Pinned Tweet

bWorker To... @bWorker... · Jun
bWorker is a defi protocol that hel
users earn up to 15% daily - 5475% APY.

Twitter

@bWorkerTW

- 11.1k followers
- Active
- New account, 7 posts in total
- Significant number of bot followers ⚠

Discord

- Not available

Telegram

@bWorkerGroup

- 31617 members
- Active mods
- Announcement Channel with 30366 members
- Significant number of bot followers ⚠

Medium

@bWorker

- 8.4k followers
- New account, 1 post in total
- Significant number of bot followers ⚠

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ **OVER 150 SUCCESSFUL CLIENTS**
- ✓ **MORE THAN 500 SCAMS EXPOSED**
- ✓ **MILLIONS SAVED IN POTENTIAL FRAUD**
- ✓ **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- ✓ **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

SPYWOLF.NETWORK

@SPYWOLFNETWORK

@SPYWOLFOFFICIAL

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.